Carnegie Mellon
Software Engineering Institute

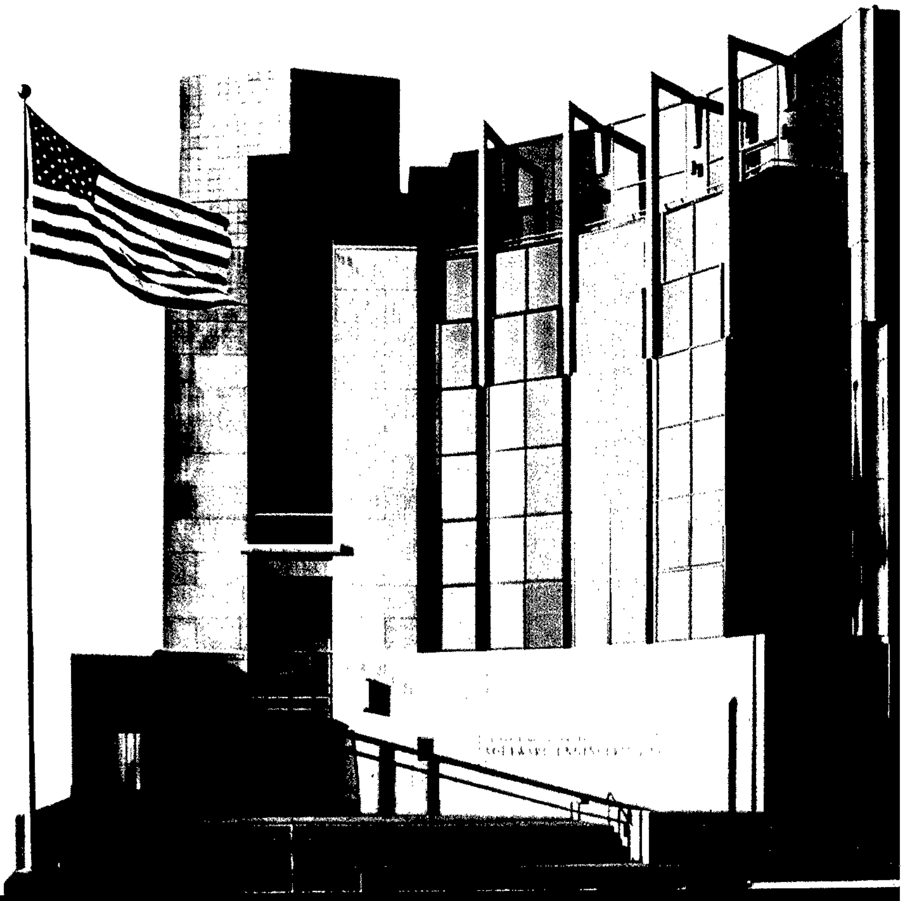
Interpreting Capability Maturity Model® Integration (CMMI®) for COTS-Based Systems

Barbara Tyson
Cecilia Albert
Lisa Brownsword

October 2003

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20031202 085



TECHNICAL REPORT
CMU/SEI-2003-TR-022
ESC-TR-2003-022

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Interpreting Capability Maturity Model® Integration (CMMI®) for COTS-Based Systems

CMU/SEI-2003-TR-022
ESC-TR-2003-022

Barbara Tyson
Cecilia Albert
Lisa Brownsword

September 2003

**Software Engineering Process Management Initiative
and
COTS-Based Systems Initiative**

Unlimited distribution subject to the copyright.

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Christos Scondras
Chief of Programs, XPK Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2003 by Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Acknowledgements

We wish to thank the National Imagery and Mapping Agency (NIMA) and the Deepwater Program of the United States Coast Guard who provided the initial funding for this work. We would particularly like to thank Mike Phillips and Jon Gross, who believed that guidance for interpreting CMMI for organizations building or maintaining solutions using off-the-shelf products was needed and would not let go of the idea until this document became a reality.

This work rests heavily on the body of knowledge formed by the members, past and present, of the Software Engineering Process Management and COTS-Based Systems Initiatives. We thank all of them for their contributions.

Abstract

Experience shows that engineering commercial off-the-shelf (COTS)-based systems requires fundamental changes from traditional engineering: adjusted roles and responsibilities, new skills, and different processes. Practitioners are often surprised to find that building and supporting COTS-based systems demands more, not less, discipline in their management and engineering practices.

Many organizations have derived benefit from process improvement using capability maturity models and want to apply them as they build COTS-based systems. In addition, organizations building COTS-based systems want to apply the Capability Maturity Model® Integration (CMMI®). This leads to the question, "How should CMMI be interpreted for organizations building, fielding, and supporting a COTS-based system?"

This report shows that developing and maintaining COTS-based systems is more than selecting products and managing vendor relationships and is, therefore, more than just applying the Supplier Sourcing discipline within CMMI. The four CMMI disciplines—Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing—require interpretation and must be used together to promote improvement of an organization's processes for developing and maintaining COTS-based systems. This report summarizes what makes COTS-based systems unique and provides high-level guidance for interpreting and using CMMI practices to facilitate appropriate processes for COTS-based systems.

® Capability Maturity Model and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

1 Introduction

1.1 Background

Using commercial off-the-shelf (COTS) products, as well as off-the-shelf products from other sources,[1] to address critical functional requirements is an increasing trend. Practical experience shows that building COTS-based systems[2] requires new skills, knowledge, and abilities; changed roles and responsibilities; and different processes from traditional product development [USAF 00]. It surprises many organizations that these new and changed processes demand more, not less, discipline in management and engineering practices.

The Capability Maturity Model® Integration (CMMI®) [CMMI 02] contains the essential elements of effective processes and provides guidance for developing processes. Many organizations have derived benefits [Goldenson 95] from process improvement using capability maturity models. Use of CMMI requires that each organization tailor and implement its management and engineering processes by interpreting the essential elements described in CMMI based on the organization's unique circumstances. These circumstances include both the character of the organization and the nature of the solutions developed and maintained by that organization.

The findings in this report are designed to provide high-level guidance on interpreting and using CMMI to facilitate the definition of appropriate processes for developers and maintainers of COTS-based systems. The analysis in this report is based on extensive knowledge from teaching and applying CMMI as well as lessons learned from over 50 COTS-based systems [CBS 02] and research by the Software Engineering Institute (SEI) COTS-Based System Initiative on processes needed in the management of COTS-based systems [Brownsword 98, OSD 00, Oberndorf 00, Carney 00, Carney 03, Albert 02], engineering techniques for designing and evolving COTS-based systems [Wallnau 02], and evaluation techniques for assessing

[1] "Other sources" include legacy systems, products found in a reuse asset library, "share" ware, open-source software, etc.

[2] This report will use the term *COTS-based system* to represent a system composed of either one substantial COTS product tailored to provide needed functionality or multiple products from a variety of off-the-shelf sources with custom products that are integrated to collectively provide the needed functionality.

® Capability Maturity Model and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

COTS-based program risks, suitability of COTS products, and appropriateness of COTS-based system designs.[3]

1.2 Target Audience

This report is designed to meet the needs of two types of organizations:

- organizations that are deriving substantial benefits through process improvement using capability maturity models and want to leverage their previous investments in process improvement to build or evolve COTS-based systems
- organizations that are building COTS-based systems and want to begin applying CMMI as their model for process improvement

1.3 Using This Report

This report is organized as follows:

- Section 2 characterizes the unique dynamics and constraints associated with the use of off-the-shelf products and summarizes the basic elements of CMMI.
- Section 3 describes how the analysis was conducted and presents general findings.
- Section 4 discusses each of the essential process demands for COTS-based systems and identifies their implications for project management, engineering, and support processes.
- Section 5 provides in-depth guidance for the CMMI project management, engineering, and support process areas as a reference for defining, tailoring, or appraising processes for COTS-based systems projects.

Sections 2 and 3 are particularly important. These sections provide the foundation for the more detailed analysis in Sections 4 and 5. Successful development and maintenance of COTS-based systems demand integrated processes that extend across all CMMI process areas. Projects that address just one of the demands, as described in Section 4, or one of the process areas, as described in Section 5, will not realize the benefits of these guidelines.

Note: This report complements the CMMI documentation [CMMI 02]. Selected information is reproduced here for ease of use. The CMMI process area descriptions, informative material, references, and glossary are used as written unless explicitly stated.

[3] In addition to the publicly available work, this document builds on *COTS-Based Systems for Program Managers* (tutorial) by L. Brownsword, P. Oberndorf, and C. Sledge; *COTS Product Evaluation* (tutorial) by P. Oberndorf, J. Dean, E. Morris, and S. Comilla-Dorda; and *COTS Usage Risk Evaluation* (CURE) by D. Carney, P. Place, and E. Morris.

2 Basics–COTS-Based Systems and CMMI

This section provides the information that is the basis of this analysis. It characterizes the unique dynamics and constraints associated with the use of off-the-shelf products and presents the fundamental process demands for developing and maintaining COTS-based systems. It also summarizes the basic elements of CMMI and describes how CMMI should be used.

2.1 Characteristics of COTS-Based Systems

In contrast with custom development, where projects can uniquely design and implement a solution to meet the demands of a particular operational environment, COTS-based systems are comprised of off-the-shelf products that are each typically developed in response to someone else's perception of needs. For example, COTS products are developed in response to each vendor's perception of the needs of a broad set of customers within the commercial marketplace.

COTS-based systems thus introduce the following new dynamics and constraints that a project's management and engineering processes must accommodate:

- Off-the-shelf products, explicitly or implicitly, make functional assumptions about the end-user's operational processes that are supported by each product. These assumptions seldom completely match the current or anticipated *enterprise processes.*[4]
- Off-the-shelf products, explicitly or implicitly, make architectural assumptions about how a given product will link to other products. To add complexity, the products often have dependencies on other specific products (or sometimes, specific versions of other products) for their successful operation.
- The supplier often retains data rights to the source code of the off-the-shelf product, and intends for the products to be used without *modification*[5] of the product. This means that developers and maintainers must frequently treat the product as a "black box" in the solution. Engineers may need to use special diagnostic techniques and tools to discover the

[4] The term *enterprise processes* is used to include the business, mission, or operational processes needed for the successful functioning of the enterprise for which the system is being developed.

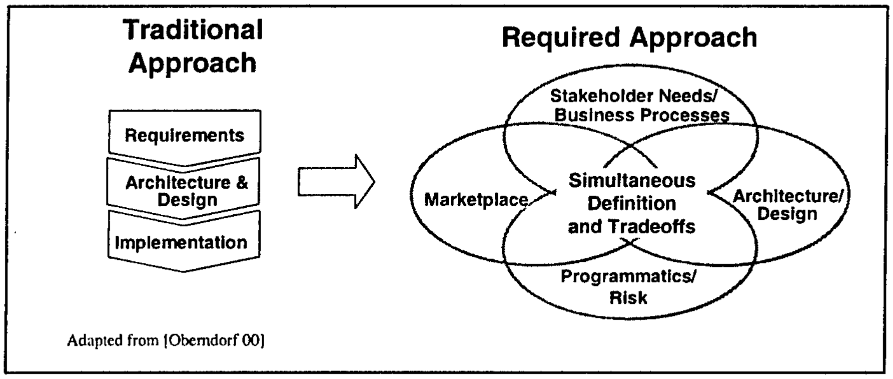
[5] The term *modification* is used to mean changes to the internals of a hardware device or the software code. This does not include vendor-provided mechanisms for tailoring the product.

implicit assumptions reflected in the product and analyze their impact on the *solution,*[6] including the enterprise processes.

- The product supplier, not any one customer, controls frequency and content of the off-the-shelf product's maintenance or improvements. For example, a commercial vendor's desire to maintain competitive advantage and the extent of competition in the marketplace will drive the frequency and content of COTS product releases.

2.2 Required COTS-Based Systems Approach

In spite of these new dynamics, many organizations have tried to use traditional, linear processes such as those shown on the left in Figure 1. These processes sequentially define the requirements, form an architecture to meet them, and then search the commercial marketplace and other sources for off-the-shelf products that fit into that architecture. These organizations rarely find this approach successful—either they cannot find products that "fit" and resort to custom development, or they try to force the products to fit by modifying or extensively tailoring the products and incur significant cost and schedule impacts with each product upgrade.



Adapted from [Oberndorf 00]

Figure 1: Fundamental Change for COTS-Based Systems

SEI experience in examining projects attempting to build COTS-based systems shows that a fundamental change is required in how COTS-based systems are engineered—and these engineering changes have attendant management and business process implications [Oberndorf 00]. As shown on the right of Figure 1, the required approach for developing COTS-based systems simultaneously defines and performs tradeoffs among four *spheres of influence* (areas of information) over the life of the solution:

- stakeholder needs and business processes - what the stakeholders want and how end users will operate using the solution

[6] The term *solution* is used to describe the integrated assembly of one or more off-the-shelf products, any required custom products (including "wrappers" or "glue"), and any changes to the enterprise processes required to successfully meet the enterprise's defined needs.

- architecture and design - what engineers can do to structure the solution architecturally so that the products work and evolve together and meet the defined needs
- programmatics and risk - what the project and end-user community can tolerate in terms of cost, schedule and risk, including management of the implementation of any needed changes to the end user's operational processes
- marketplace - what current and emerging products, technologies, and standards are available and how the products are likely to change over the life of the solution

While the first three spheres listed above have analogues in custom development processes, the marketplace is a potent addition. Forming a COTS-based system solution requires that information be simultaneously collected and updated from each sphere at approximately the same level of detail, since information in one sphere often depends on and identifies the information required from another sphere. As the information among the spheres is analyzed, mismatches are identified and resolved through negotiation with the affected stakeholders. In practice, this drives the process activities to "gather a little," "analyze and negotiate a little," and then gather some more, and analyze and negotiate further.

In addition, each product used in the solution has a maintenance and improvement cycle that is independent of the project needs and of the maintenance and improvement cycles of any other products used in the solution. Due to this dynamic, projects will need to simultaneously gather information, analyze the impact of the new information on the solution, and negotiate any mismatches among the affected stakeholders repeatedly throughout the life of the solution.

2.3 Process Demands for COTS-Based Systems

The new dynamics and required COTS-based system approach described in the previous subsections demand a number of essential changes to the management and engineering processes needed to build, field, and support COTS-based systems:

- The simultaneous definition and trades of the spheres of influence must continue throughout the life of the solution. A process for COTS-based systems is ultimately an act of reconciling the diverse expectations of stakeholders with a current understanding of the capabilities of available COTS products. The balance among the frequently competing expectations of the stakeholders representing each of the spheres of influence must be maintained throughout all phases of the project.
- The engineering of enterprise processes must occur concurrently and in coordination with the engineering of the solution. Because COTS products implement assumptions about the enterprise processes, the end-user community must be willing and able to modify its enterprise processes to align with those assumptions.

- The formation of requirements depends on an understanding of the opportunities and limitations of the available off-the-shelf products. For COTS-based systems, it is folly to attempt to define a detailed, fixed set of requirements first and then develop a solution that satisfies them. Rather, requirements emerge and are adjusted as different off-the-shelf products and combinations of products are explored and the definition of the solution is refined.
- The marketplace is monitored continuously across the life of the solution. Given the volatility of the marketplace, processes for a COTS-based system must anticipate and track changes in the off-the-shelf products that could affect the solution. Indicators of these changes will be provided through tracking relevant market segments, including market share distribution; understanding the needs of other customers; and monitoring current and emerging products and technologies.
- A flexible architecture is developed early and maintained throughout the life of the solution. Since the products in the solution are "owned" by their suppliers, an architecture that can retain its structure and cohesiveness, yet allow the solution to respond easily to product changes, becomes a key organizational asset.
- Disciplined spiral or iterative practices, with frequent executable representations of the solution, are effectively implemented. Spiral development practices allow the discovery of the critical attributes of the solution through an evolutionary exploration of the highest risk elements [Boehm 00]. Executable representations that characterize the enterprise processes and simultaneously mature the architecture facilitate direct feedback from all of the affected stakeholders.
- Stakeholders are directly and actively involved throughout the life of the solution. Active, often day-to-day participation by the affected stakeholders will be necessary as the definition of the solution evolves. This participation will also facilitate the timely resolution of mismatches as they are discovered.

Section 4 will further discuss each of these process demands with their process implications. For readers with limited familiarity of CMMI, a brief summary is provided in the following subsection.

2.4 Attributes of CMMI

CMMI integrates proven approaches for process improvement, organizational change, and system and software engineering to help an organization improve the processes it uses to develop and maintain software-intensive systems. To facilitate process improvement, CMMI helps an organization examine its current processes; establish priorities for improvement of those processes; and implement these improvements across the organization.

Process areas are the primary building blocks of CMMI. Process areas are not processes nor are they process descriptions. Rather, process areas describe critical elements of successful processes. The essential elements of each process area are represented in the set of goals that must be met by successful processes. In addition, each process area is made up of practices that, when performed, satisfy the process area goals.

CMMI contains process areas that comprise four *disciplines*—Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing.

- System Engineering focuses on transforming customer needs, expectations, and constraints into product solutions and supporting these product solutions throughout the life of the product.
- Software Engineering focuses on applying systematic, disciplined, and quantifiable approaches to the development, operation, and maintenance of software.
- Integrated Product and Process Development (IPPD) provides a systematic approach that achieves a timely collaboration of relevant stakeholders throughout the life of the product to better satisfy customer needs, expectations, and requirements.
- Supplier Sourcing covers the enhanced source analysis and monitoring of supplier activities needed when the project acquires critical products from suppliers who perform functions or add modifications to products that are specifically needed by the project.

CMMI is available in two representations: staged or continuous. The continuous representation groups process areas by affinity categories and designates capability levels for process improvement within each process area. The staged representation organizes process areas into five maturity levels to support and guide process improvement. Organizations must select the representation that best fits the organization's process improvement needs.

In the continuous representation, the basis for this analysis, process areas are grouped into four affinity *categories*—Process Management, Project Management, Engineering, and Support. The four categories with their associated process areas are noted in Table 1.

Table 1: CMMI Process Areas by Affinity Category

Category	Process Areas
Process Management process areas provide the organization with a basic capability to document and share best practices, organizational process assets, and learning across the organization.	Organizational Process Focus Organizational Process Definition Organizational Training Organizational Process Performance Organizational Innovation and Deployment
Project Management process areas address the basic activities related to establishing and maintaining the project plan, establishing and making commitments, monitoring progress against the plan, taking corrective action, and managing supplier agreements.	Project Planning Project Monitoring and Control Supplier Agreement Management Integrated Project Management for IPPD Risk Management Integrated Teaming Integrated Supplier Management Quantitative Project Management

Engineering process areas cover the development and maintenance activities that are shared across engineering disciplines.	Requirements Management Requirements Development Technical Solution Product Integration Verification Validation
Support process areas address basic support functions that are used by all process areas.	Configuration Management Process and Product Quality Assurance Measurement and Analysis Decision Analysis and Resolution Organizational Environment for Integration Causal Analysis and Resolution

2.5 Using CMMI

CMMI is not intended to be prescriptive or to define how to do software development. Rather, CMMI provides the essential elements of effective processes to be used by organizations when improving their own processes.

Each organization must use professional judgment to interpret the CMMI practices. Although process areas depict behavior that any organization should exhibit, practices must be interpreted using an in-depth knowledge of the CMMI model, the organization, the business environment, and the specific circumstances involved.

To interpret the model's practices, it is important to consider the overall context in which they are used and determine how well the practices satisfy the goals of a process area within that context. CMMI models do not imply which processes are right for a given organization or project. Instead, CMMI models establish criteria necessary to plan and implement processes selected by the organization for improvement based on business objectives.

3 Using CMMI Models for COTS-Based Systems

This section describes the approach used for this analysis and provides a summary of its global findings. Subsequent sections will build upon and amplify these global findings to provide practical advice in defining a project's processes for developing and maintaining a COTS-based system (Section 4) and interpretive guidance for using each of the CMMI process areas (Section 5).

3.1 Summary of the Analysis Approach

The analysis reflected in this report is based on integrating the experiences of the Software Engineering Process Management and COTS-Based Systems Initiatives.

The continuous representation of CMMI [CMMI 02] was used as the basis for this analysis. This representation was selected because it supports selecting process areas that support building, fielding, and supporting a COTS-based system without consideration of the maturity level required to implement any given process area. For example, this report will show that the practices of the Integrated Supplier Management process area are essential to a COTS-based system development—even though the staged representation characterizes this process area as Maturity Level 3. The findings in this report, however, are equally applicable to organizations improving their processes using the staged representation. These organizations must still focus on meeting the current needs of projects that are developing or maintaining COTS-based systems. This may mean using practices more typically found at higher maturity levels—albeit in a way that is consistent with the abilities of the members of each of the affected projects.

The Evolutionary Process for Integrating COTS-Based Systems (EPIC) [Albert 02] was used as an exemplar of a process for building, fielding and supporting a COTS-based system. EPIC embodies the lessons learned and research of the COTS-Based Systems Initiative by defining a structured flow of key activities and artifacts that implement the management and engineering practices necessary to more effectively leverage the use of off-the-shelf products. EPIC emphasizes concurrent discovery and negotiation of the diverse spheres of influence discussed in Section 2.2 of this report using a risk-based, disciplined spiral engineering approach.

The process areas in all four disciplines of CMMI were analyzed in the context of the practices implemented in EPIC. The analysis included the specific goals, specific practices, and informative material for the process areas in each of the four CMMI disciplines. The Project Management, Engineering, and Support process area categories were emphasized since these process areas most directly affect a project developing or maintaining a COTS-based system.

Because the generic goals and generic practices focus on the institutionalization of the processes, they were not included in this analysis. These goals and practices should be applied regardless of the type of project.

3.2 General Findings

The analysis confirmed that CMMI provides a sound foundation for developing and improving COTS-based systems processes. However, contrary to the view held by some, developing and maintaining a COTS-based system requires more than just applying the Supplier Sourcing discipline:

1. Each of the CMMI disciplines was found to be critical in building, fielding, and supporting a COTS-based system.
    - The *Systems Engineering* and *Software Engineering* disciplines together provide a basis for defining the management and engineering processes necessary to develop and maintain COTS-based systems.
    - *IPPD* provides an approach for the timely involvement of affected stakeholders in support of the negotiations among the spheres of influence.
    - *Supplier Sourcing* is very relevant for selecting and managing suppliers of custom products. For COTS-based systems, however, this discipline provides some, but not all, of the necessary practices for developing appropriate relationships with suppliers of off-the-shelf products.
2. All of the CMMI process area categories are important to a COTS-based system project.
    - The *Project Management* category addresses the basic activities related to establishing and maintaining the project plan for a COTS-based system, establishing and maintaining commitments among all affected stakeholders, monitoring progress against the plan, and establishing and managing appropriate relationships with suppliers.
    - The *Engineering* category addresses the basic activities related to identifying and managing the evolving definition of the requirements for a COTS-based system, developing, verifying, and validating technical work packages that characterize what is known about the solution at any point in time, and fre-

quently integrating executable representations that demonstrate that the evolving definition of the solution is both useful and buildable.

- The *Process Management* categories provide the organization with the capability to document and share best practices, organizational process assets, and learning across the organization. While this analysis shows no COTS-based system specific interpretations for this category, for organizations just starting to build or support COTS-based systems, the need to contribute COTS-based system process assets to the organization's process asset library may be particularly relevant.
- The *Support* category addresses activities that are used in the context of performing other processes.

3. CMMI Process areas in the Program Management, Engineering, and Support categories will need to be integrated and applied differently to accommodate the process demands of COTS-based systems:

    - The simultaneous definition and trades among the spheres of influence demands the concurrent definition, verification, and validation of requirements and the technical solution—to include the evaluation and selection of appropriate off-the-shelf products.
    - The activities that engineer and implement enterprise processes (an area not directly covered in CMMI) must occur concurrently and in coordination with the activities that engineer and implement the technical solution.
    - The formation of requirements must be based on a clear understanding of the opportunities and limitations of the available off-the-shelf products and an understanding of the viability of candidate technical solutions that use the products.
    - The marketplace must be monitored continuously across the life of the solution for indicators of potential changes to the off-the-shelf products that could affect the solution and for opportunities to influence those changes.
    - The alternate architectures must be sufficiently flexible to support the use and evolution of candidate off-the-shelf products. These architectures must be developed early and maintained throughout the life of the solution.
    - The project's management and engineering activities must be aligned to accommodate disciplined spiral or iterative practices that respond directly to the highest priority project and solution risks. Frequent executable representations of the solution must be used to validate stakeholder consensus and progress in the evolving definition of the solution.
    - All affected stakeholders, who will likely represent very diverse interests, must be directly and actively involved in engineering as well as management activities throughout the life of the solution.

4. The integration of activities that plan, design, and deploy any necessary changes to the enterprise processes is critical to developing and maintaining a COTS-based system. While the need for this integration is briefly mentioned in the Technical Solution process area, no specific practices exist in CMMI to explicitly address these activities. The process areas of the Process Management category, although focused on the improvements of an organization responsible for the development or maintenance of a system—not the end-user community—provide valuable insights to the practices necessary to make changes to the enterprise processes.

The findings in this report are presented in three major groupings. This subsection addressed the general findings. Section 4 will amplify these findings in the context of the COTS-based system process demands addressed in Section 2.3. Section 5 will amplify these findings in the context of each of the CMMI project management, engineering, and support process areas.

4 Meeting the Demands of COTS-Based Systems

The following subsections present our guidance from a COTS-based system process perspective. The subsections are organized by the essential process demands identified for COTS-based systems discussed in Section 2 and listed in Table 2.

Table 2: COTS-Based System Essential Process Demands

The simultaneous definition and trades of the spheres of influence must continue throughout the life of the solution.
The engineering of enterprise processes must occur concurrently and in coordination with the engineering of the solution.
The formation of requirements is based on a clear understanding of the opportunities and limitations of the available off-the-shelf products.
The marketplace is monitored continuously across the life of the solution.
A flexible architecture is developed early and maintained throughout the life of the solution.
Disciplined spiral or iterative practices, with frequent executable representations of the solution, are effectively implemented.
Stakeholders are directly and actively involved throughout the life of the solution.

For each essential process demand, the demand on the process is discussed; the implications for project management, engineering, and support processes are identified; and highlights of the guidance for the most affected CMMI process areas are provided. To aid in understanding this guidance, the affected attribute of the CMMI process area is identified (present tense) and the interpretation that will be required for COTS-based systems (future tense) follows. So they can be readily identified, CMMI process area names are indicated in the text with **bold** font. Not all affected CMMI process areas are discussed in this section. Additional information can be found for all process areas in Section 5.

4.1 Support Simultaneous Definition and Trades

Where many projects view each process area sequentially—developing requirements, designing a technical solution, integrating the product, then verifying and validating it—successful development of a COTS-based system requires a project to fully integrate all process areas so that the requirements and the solution evolve together as end users and other affected stakeholders mature their understanding of their needs and the capabilities in the marketplace and other sources.

Implications for project management processes:

- The **Project Planning** practices establish and maintain plans that define the project activities. To accommodate the required simultaneous definition and trades, however, project plans for a COTS-based system must orchestrate the execution of all engineering process areas concurrently with extensive interaction among them.
- The **Risk Management** practices establish mechanisms to continuously identify and analyze risks across a project. For a COTS-based system project, these practices can be used to ensure that an appropriate balance between the spheres of influence is maintained. Note that for a COTS-based system project, a failure in the simultaneous definition and trades, such as allowing one of the spheres of influence to be defined ahead of the other spheres, will pose significant risk.

Implications for engineering processes:

- For COTS-based systems, the chief effect is not on any single process area, but in the *interdependence* of the respective process areas that must be combined and related in the project's processes. For example, effectively implementing **Requirements Development** practices will require the following: an understanding of the relevant capabilities of the marketplace as described in **Integrated Supplier Management** practices; the end-user community's agreement to make any needed changes to its enterprise processes to accommodate candidate off-the-shelf products;[7] formation of an architecture and design as described in **Technical Solution** practices; and updated associated cost, schedule, and risk as described in **Project Planning** practices.
- The engineering processes for COTS-based systems projects must expect changes from the marketplace—and these changes can occur during any phase of the life cycle. Project commitments reflected in plans, requirements, architectures, designs, and enterprise processes will need to be revisited to accommodate these changes.
- As the information in the four spheres of influence expands in a COTS-based system, **Verification** practices can be used to determine that the information in each sphere is sufficient and complete for the current level of maturity of the definition of the solution.

[7] As previously note in Section 3.2, the Process Management process areas can be adapted to effect the necessary changes to the enterprise processes.

Similarly, **Validation** practices can be used to determine that the solution described will meet real enterprise needs.

Implications for support processes:

- The **Decision Analysis and Resolution** practices establish guidelines to analyze possible decisions using a formal evaluation process. For a COTS-based system, these practices can be used to determine the issues subject to various levels of review and the processes by which those reviews are conducted.

4.2 Coordinate Enterprise Process Engineering with System Engineering

In custom development, the solution is specifically designed to meet the demands of previously defined enterprise processes. In a COTS-based system, the situation is very different. In this case, the enterprise processes must be determined and negotiated as the capabilities of the solution are defined. This is made more complex as off-the-shelf product upgrades become available throughout the life of the solution and force the enterprise processes to be redetermined and renegotiated as part of evaluating the product release.

Implications for project management processes:

- CMMI does not directly address managing changes to the enterprise processes. However, the Process Management process areas provide considerable insight into the practices that may be necessary and can be adapted to meet this need. For a COTS-based system, changes to enterprise processes to better align with the capabilities of the off-the-shelf products must be explicitly planned and implemented as part of the project.

Implications for engineering processes:

- The development and analysis of alternative solutions is described in practices in **Technical Solution**. For a COTS-based system, these practices should explicitly include identification and analysis of the affected enterprise processes. Both improvement opportunities and impacts on current enterprise processes need to be considered as part of the selection of the preferred solution and its off-the-shelf products.

Implications for support processes:

- The practices in the IPPD process areas provide the mechanisms for enabling the disparate stakeholder groups to work effectively together. In particular, practices in **Organizational Environment for Integration** are key enablers in establishing a vision for the organization that is shared among the stakeholders and in developing a set of appropriate incentives that motivate the integrated behaviors necessary. For a COTS-based system, these practices will be essential to coordinating the development and enterprise engineering as the business and operational processes are negotiated as part of the development process.

4.3 Form Requirements with Knowledge of Off-the-Shelf Products

For a custom development, the inclination is to define a detailed set of requirements at the start of a project and to attempt to keep those requirements stable to define the architecture and resulting solution implementations. Two conditions impinge on this view in a COTS-based system. First, end users will come to better understand the product capabilities by interacting with the products and can modify their requirements to better leverage the use of those products. Second, the marketplace is continually introducing new technology and capabilities. Requirements will need to be reevaluated, with adjustments as appropriate, across the life of the solution.

As enterprise processes and end-user requirements are modified to accommodate the use of off-the-shelf products, the implementation of the agreed-upon enterprise process changes must be explicitly coordinated with the development of the solution and managed as part of the project. Depending on the volatility of the selected products and the dynamics of the organization, the definition and implementation of new requirements and enterprise processes may become continuous activities. Managing these changes will require disciplined, integrated, and mature management processes.

Implications for project management processes:

- For COTS-based systems, practices in **Project Planning** must be used to continually manage the evolution of the project's plans in such a way as to encourage and reinforce the continual discovery of requirements. This will require striking a balance between the need for adapting the requirements as more is learned about the opportunities and limitations in off-the-shelf products and the need for requirements to be stable enough to field useful capability in a timely manner. Spiral development practices for planning and managing projects are particularly germane.

Implications for engineering processes:

- The practices in **Requirements Development** produce and analyze customer, product, and product-component requirements. For a COTS-based system, these practices must be performed simultaneously with those in **Technical Solution**, **Product Integration**, **Verification**, and **Validation**, using a robust enterprise process engineering approach. Whenever possible, requirements should be modified to align with the capabilities of the candidate off-the-shelf products. To provide the necessary flexibility, requirements must be kept fluid until what stakeholders want is balanced with what can be obtained from the marketplace, assembled by the engineers, and supported by project management. Prioritizing the requirements with an emphasis on identifying a minimum set of "must have" stakeholder needs is a useful mechanism. In addition, expressing requirements in the form of operational scenarios or use cases that emphasize what the solution has to do rather than how the solution must be implemented is particularly useful in understanding the impacts of the trades.

- Practices from **Requirements Management** that manage the requirements of the project's products will be required from the start of a COTS-based system project—earlier than many projects currently apply them. Note that for a COTS-based system, "managing" requirements means that as potential changes are identified, the impact of the change is determined, and negotiated and approved changes are tracked and managed. Experience has shown that robust requirements management processes are required to accommodate the limitations and volatility of the products in a COTS-based system.

4.4 Monitor the Marketplace Continuously

COTS-based system projects need processes that proactively identify and evaluate off-the-shelf products and potential changes to already selected products. Projects must analyze the impact of each product on and its benefits to the current understanding of the solution, and determine if, how, and when to accommodate the changes. However, knowledge of each product's current capabilities is not enough. When a product is selected, the project inherits the supplier's maintenance strategy and the potential for disruption caused by "improvements" made to the product over the life of the solution. It is also important to know who else is using the product and how they are using it. In addition, an understanding of the product's underlying technologies, how these technologies are changing, and how those changes could affect the solution is required.

Knowledge of the marketplace is needed not just once at the start of a project, but must be kept current until the solution is retired or replaced. Effective relationships with suppliers, other customers, and participation in user groups and standards bodies are key sources of marketplace information and provide valuable opportunities to influence product directions.

Note: Suppliers of off-the-shelf products will seldom respond to direction as a developer of custom software would. Nor will all suppliers encourage (or allow) a close working relationship. The nature of the relationship will depend on the importance of the product to the solution and the importance of the project to the supplier.

Implications for project management processes:

- **Integrated Supplier Management** defines practices that establish a foundation for developing appropriate supplier relationships. However, the specific practices tend to emphasize relationships with suppliers of custom products, not suppliers of off-the-shelf products. To build processes appropriate for COTS-based systems, the specific practices must be reoriented. For example, these suppliers of off-the-shelf products will not generally allow monitoring of their processes or work products. Hands-on evaluation of each off-the-shelf product release (including any patches) is more realistic.

- The **Project Planning** practices establish mechanisms to plan for the activities and resources needed for monitoring the marketplace as an integral part of planning the project. For a COTS-based system, planning processes must include the resources (budget, time, facilities, and skilled personnel) necessary to support the monitoring of the marketplace and product evaluation across the life of the solution. In particular, an experimentation facility that allows for hands-on examination of the products under consideration will be required for the life of the solution.

Implications for engineering processes:

- The **Integrated Supplier Management** practices, though not typically considered engineering practices, are useful for identifying and evaluating potential sources of products and for the ongoing monitoring of changes in the marketplace. However, these practices should not be used to select products. Product selection must be integral to developing and analyzing alternative architectures as described in **Technical Solution** practices.
- For COTS-based systems, identification and analysis of the impact of marketplace changes must be considered in both the **Technical Solution** and **Requirements Development** process areas as well as in engineering enterprise processes. Engineering processes should include activities that create and maintain the experimentation or test bed facilities so that engineers and end users can conduct hands-on evaluations of off-the-shelf products.

Implications for support processes:

- The practices of **Process and Product Quality Assurance** must be viewed differently for an off-the-shelf product. Since these products were not developed to directly meet the needs of the project, the work products for these products may not conform to the standards set for the project. Diagnostic techniques will be required to ensure sufficient insight into each product's quality and to understand how that quality will affect the quality of the delivered solution.

4.5 Develop Early and Maintain a Flexible Architecture

Since the off-the-shelf products are "owned" by their vendors or suppliers, the framework by which the products are combined with other products of the solution to provide desired functionality—the architecture—becomes an important strategic asset. Further, the *flexibility* and *evolvability* of a COTS-based system become critical quality attributes of the solution. The architecture is evolvable if it can retain its structure and cohesiveness yet allow the solution to respond efficiently to product upgrades, technology advances, and new operational or business needs.

Implications for project management processes:

- **Project Planning** practices describe the mechanisms necessary to manage the architecture as a corporate asset. For a COTS-based system, planning must account for the resources necessary to form and evaluate, through one or more executable representations, the flexibility and evolvability characteristics of the solution and its architecture.

- For COTS-based systems, the architect of the solution must have excellent current and predictive knowledge of the domain and marketplace products and technologies. The uncertain availability of this caliber of engineer may represent a substantial risk to the project. Planning for the project must include identifying and scheduling personnel with the necessary skills. In addition, processes that manage risk should identify and actively develop mitigations for this risk.

Implications for engineering processes:

- The flexibility and evolvability attributes are so important to a COTS-based system that **Verification** and **Validation** practices should be used to assess these properties in the architecture as demonstrated in simulations and prototypes as the solution's definition evolves.
- **Technical Solution** practices indicate that "make-or-buy" analysis begins early in the project; it also indicates completion of that analysis when a decision is made. For COTS-based systems, this analysis will not be complete until the solution is retired. Previous decisions may have to be revisited when an existing product changes or a new product becomes available.
- A strategy that describes project standards or protocols that will be used to link off-the-shelf products and other elements of the solution must be included as an artifact of **Technical Solution** practices for a COTS-based system. Individual strategies must be developed to meet the needs of each defined alternative. Each strategy must also describe the level of effort required to create and maintain the "wrappers" or "glue" as the products change across the life of the solution.

4.6 Implement Disciplined Spiral Practices with Frequent Executables

Spiral development allows the discovery of the critical attributes of the solution through an evolutionary exploration of the highest risk elements of the solution and the COTS products available to address them. It also facilitates the frequent and direct feedback from all affected stakeholders and reduces the risks due to misunderstandings or unforeseen technical and operational factors through evolving executable representations of the solution. The executable representations must be sufficient to demonstrate how the enterprise processes are affected by the solution.

Implications for project management processes:

- If a spiral development process has not been implemented, COTS-based system projects may need to revamp their planning processes to align with this approach. In spiral development, planning evolves rather than being "done once" with updates only based on major, frequently unforeseeable, changes. **Project Planning** practices establish mechanisms that can be used to continuously update the project plan based on what is currently known of project and solution risks. For a spiral development approach, the project plan is updated in each iteration[8] as detailed planning for the iteration reflects the lessons learned and risks mitigated in preceding iterations. This results in an evolving definition of project objectives as the project's risk profile changes.

- The practices of the **Risk Management** process area are essential to the successful management of a COTS-based system project. In a spiral development approach, prioritized risks are used to directly plan and manage the project—and the plan is updated as known risks are suitably mitigated and new risks are identified. The objectives for each iteration are designed to mitigate the highest priority remaining project risks. The management of risk must be an integral part of project planning, project monitoring, and engineering activities.

Implications for engineering processes:

- To mitigate risk in a COTS-based system project, requirements must be explored and negotiated, alternative designs must be created and evaluated, and executable representations of the solution must be assembled. Executable representations need to be built in each iteration to demonstrate consensus and progress. Therefore, for a COTS-based system, the practices in **Requirements Development**, **Technical Solution**, and **Product Integration** will take place in *every* iteration.
- In spiral development, the solution, any products (including off-the-shelf products), and selected intermediate work products require almost continuous re-verification and re-validation as the definition of the solution evolves. The **Verification** and **Validation** process areas can be valuable in defining the processes necessary in this area. In particular, specific practices that concurrently develop verification procedures and criteria with the solution design will be important in defining the activities for each iteration.

Implications for support processes:

- The processes that implement the **Configuration Management** practices must be very robust for a COTS-based system. The definition of what to control and when it will be "baselined" has to be carefully considered. Where in a more traditional approach, selected artifacts are baselined at specific project milestones, a project using a spiral development approach must develop the engineering artifacts concurrently. Definition and control of these artifacts will begin very early in the development process and continue as the definition of the solution evolves.
- While much work has been done to measure progress and quality of a custom software development, the same cannot be said for solutions that depend on off-the-shelf products. This makes the practices of **Measurement and Analysis** more critical, as each organization will have to establish measurement objectives, measurements, and procedures based directly on its own business goals.

[8] Also referred to as a *spiral*.

4.7 Involve Stakeholders Directly and Actively

The required level of commitment throughout the life of the solution is significant, even unprecedented, particularly for the end-user community. However, continuous participation from all affected stakeholders is essential to the success of the project because the activities that identify, evaluate, and select COTS products will shape the end-user operational processes and define the functionality, cost, and schedule for the solution that will be delivered. Affected stakeholders must be available to quickly resolve mismatches as they are discovered between the available COTS products, and to concurrently agree that the evolving definition of the solution will meet their needs.

Implications for project management processes:

- The **Integrated Project Management for IPPD, Integrated Teaming,** and **Organizational Environment for Integration** process areas provide the necessary foundation to form and sustain empowered teams. The IPPD process areas will provide the basic mechanisms needed for defining and managing the stakeholder involvement necessary to support the development of COTS-based systems. The broad and possibly disparate group of stakeholders affected by a COTS-based system will make the practices defined in the IPPD discipline particularly important—as well as the discipline amplifications for IPPD throughout all of the process areas.

- Applying **Project Planning** practices in a COTS-based system project will ensure that the necessary resources for stakeholder involvement are included. Planning for the project will need to include ensuring that relevant stakeholders are involved in requirements negotiations and that adequate resources are available to support hands-on exploration of candidate COTS products in an operationally applicable context.

Implications for engineering processes:

- While the **Validation** process area indicates that "often" end users are involved, for COTS-based systems, the end users will *always* need to be involved in validation to confirm the results of any and all negotiations.

Implications for support processes:

- **Organizational Environment for Integration** practices provide a necessary foundation to form and sustain empowered teams. In the most effective COTS-based systems projects, the solution development team must explicitly include the suppliers of the off-the-shelf products as well as the more traditional engineers, end-users, customers, testers, maintainers, and s.o on.

5 Interpreting CMMI Process Areas

This section provides guidance for each CMMI project management, engineering, and support process area. This guidance is designed to be a reference in defining, tailoring, or appraising processes for COTS-based systems projects. For convenience, the purpose, specific goals, and specific practices for each CMMI process area are provided in text boxes.

5.1 Project Management Process Areas

The CMMI project management process areas cover the management activities related to planning, monitoring, and controlling the project. As with a custom development project, there is a critical need for these management activities for projects producing COTS-based systems. For a COTS-based system, however, there are some unique considerations. These considerations and their implications are summarized in Table 3 and will be discussed in the following pages.

Table 3: Summary of Implications for Project Management Process Areas

Process Area	Interpretation
Project Planning	Orient for spiral development, if not already implemented. Identify affected stakeholders and obtain needed commitments for their active involvement. Add COTS-based system unique budget and facility needs.
Project Monitoring and Control	Use as-is—but new activities, metrics must be monitored.
Supplier Agreement Management	Use as-is for contracts with suppliers of custom products; reorient for licenses with suppliers of off-the-shelf products. Reviewing and selecting COTS products must be included in engineering processes.
Integrated Project Management for IPPD	Use as-is—it is particularly important to continuously involve stakeholders in solution definition and involve affected stakeholders in negotiations and trades.
Risk Management	Use risks to plan and manage all project/iteration activities. Add COTS-based system unique risk areas.
Integrated Teaming	Develop techniques to quickly involve very diverse stakeholders and sustain that involvement throughout the project.
Integrated Supplier Management	Use as-is for suppliers of custom products; reorient for suppliers of off-the-shelf products and market research. Develop ways to stay current in product directions and to influence directions of key off-the-shelf products through relationships with suppliers, other customers, and standards bodies. Monitoring/evaluating supplier's processes may not apply.
Quantitative Project Management	Use as-is.

Project Planning

Purpose: Establish and maintain plans that define project activities.

If a spiral development approach is not already implemented, planning processes may need to be revamped. In this approach, planning is a continuous activity that responds directly to newly discovered or mitigated risks and to what is learned about the solution as the development proceeds.

SG 1: Estimates of project planning parameters are established and maintained.

SP 1.1-1: Establish a top-level work breakdown structure (WBS) to estimate the scope of the project.

SP 1.2-1: Establish and maintain estimates of the attributes of the work products and tasks.

SP 1.3-1: Define the project life-cycle phases upon which to scope the planning effort.

SP 1.4-1: Estimate the project effort and cost for the work products and tasks based on estimation rationale.

In addition to any changes resulting from spiral development, the use of off-the-shelf products introduces new cost elements. Estimation and tracking of costs, facilities, and personnel must accommodate new and changed elements. For example, the risk that disruptive changes in the off-the-shelf products will affect the solution development plans or the enterprise processes means that relationships with vendors, users groups, and standards bodies will be required to continuously gather enough information about market drivers to anticipate the impact of product changes on the evolving definition of the solution. In addition, the required direct stakeholder involvement from the customer and end users in the planning process may impact ongoing enterprise functions and require additional resources.

Technical Solution practices expect that alternative solutions will be formed. There will likely be different project planning estimates for each alternative. These estimates will be fuzzy early on and will be refined for the most attractive alternatives as more is learned about the stakeholders' needs, the available off-the-shelf products, and the effort that will be required to build an acceptable solution using those products.

SG 2: A project plan is established and maintained as the basis for managing the project.

SP 2.1-1: Establish and maintain the project's budget and schedule.

SP 2.2-1: Identify and analyze project risks.

SP 2.3-1: Plan for the management of project data.

SP 2.4-1: Plan for necessary resources to perform the project.

SP 2.5-1: Plan for knowledge and skills needed to perform the project.

SP 2.6-1: Plan the involvement of identified stakeholders.

SP 2.7-1: Establish and maintain the overall project plan content.

To accommodate spiral development, the project plan tends to be very high level. It will describe, at a coarse-grained level: the capability that will be built and fielded; the key milestones and resources needed to deliver the capability; the number of iterations required to mitigate risk; and the allocation of the known risks to each iteration.

Detailed planning consists of a fine-grained plan for the current iteration. The iteration plan defines the goals and objectives for the iteration and determines the resources (including cost and schedule) necessary to meet them. The objectives for any given iteration are designed to meet high-priority functional needs and to mitigate high-priority risks to the project. (Note: The greatest risk posed to the project may be implementing the changes to the enterprise processes to match those in the available products.) Practitioners have found it useful to constrain each iteration to the amount of work that can be done in a fixed, relatively short period of time (one to eight weeks).

All work in a project is conducted through one or more iterations—including the work of building and maintaining the project plan. Later iterations build on the results of earlier iterations to produce the desired capability. The project plan is updated as each iteration is planned. Therefore, it always reflects what is known about the project needs and a current assessment of project risks.

SG 3: Commitments to the project plan are established and maintained.

- SP 3.1-1: Review all plans that affect the project to understand project commitments.
- SP 3.2-1: Reconcile the project plan to reflect available and estimated resources.
- SP 3.3-1: Obtain commitment from relevant stakeholders responsible for performing and supporting plan execution.

In many projects, the commitments are relatively stable. In a COTS-based system, the plans are flexible and will change in reaction to many forces inside and outside of the project. The commitments to support the updated plans have to keep pace with these changes and may need to be renegotiated with each iteration.

Project Monitoring and Control

Purpose: Provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.

Monitoring and controlling a COTS-based system project is challenging in that, as noted in discussion of **Project Planning**, the project is re-planned in each iteration based on an updated understanding of the remaining risks. In addition, the volatility of the off-the-shelf products will make monitoring and controlling the project difficult. New releases of selected products may add or delete features that will affect the use of that product in the solution. These changes, driven by forces outside the control of the project, may force the reconsideration of decisions that had already been made and cause the project to reverify and revalidate work that had been considered completed.

SG 1: Actual performance and progress of the project are monitored against the project plan.

- SP 1.1-1: Monitor the actual values of the project planning parameters against the project plan.
- SP 1.2-1: Monitor commitments against those identified in the plan.
- SP 1.3-1: Monitor risks against those identified in the project plan.
- SP 1.4-1: Monitor the management of project data against the project plan.
- SP 1.5-1: Monitor stakeholder involvement against the project plan.
- SP 1.6-1: Periodically review the project's progress, performance, and issues.
- SP 1.7-1: Review the accomplishments and results of the project at selected project milestones.

In a custom development project, it may be possible to merely track progress against a predefined project plan. In a COTS-based system project, however, the project plan is updated in each iteration. It is important to monitor the progress of the project with an emphasis on what was learned during the iteration and what those lessons mean to the project's risk profile.

SG 2: Corrective actions are managed to closure when the project's performance or results deviate significantly from the plan.

- SP 2.1-1: Collect and analyze the issues and determine the corrective actions necessary to address the issues.
- SP 2.2-1: Take corrective action on identified issues.
- SP 2.3-1: Manage corrective actions to closure.

In a risk-based spiral development approach, deviations from the project plan are handled in the context of the project's risk profile and risks are assigned for mitigation to one or more iterations in the high-level project plan. It may be many iterations before actions are taken in response to any single identified issue.

Supplier Agreement Management

Purpose: Manage the acquisition of products from suppliers for which there exists a formal agreement.

As written in CMMI, this process area focuses on the mechanisms needed for developing and managing formal purchasing and contractual agreements with suppliers—emphasizing suppliers with whom there is a contractual relationship. While the specific goals and practices are useful, suppliers of off-the-shelf products will seldom react to specific program needs as suppliers contracted to perform custom development would. Further, licenses grant permission to use a specified product under defined conditions and typically don't commit the supplier to meet unique project requirements—or even promise quality or functional behavior of the product.

SG 1: Agreements with the suppliers are established and maintained.

- SP 1.1-1: Determine the type of acquisition for each product or product component to be acquired.
- SP 1.2-1: Select suppliers based on an evaluation of their ability to meet the specified requirements and established criteria.
- SP 1.3-1: Establish and maintain formal agreements with the supplier.

In a COTS-based system, emphasis is placed on selecting the *product* that will best meet stakeholder needs—an engineering activity—rather than selecting a qualified supplier—a management activity. Confidence that the supplier can or will support the product will be evaluated as a factor in selecting the product. The formal agreement with the supplier of a selected product may be a license that grants the project permission to use the product within predefined conditions. The license should be negotiated to align as closely as possible with project and enterprise needs. It will be important that the project understand the typical (not always standard) license agreements made in the relevant segment of the marketplace.

SG 2: Agreements with suppliers are satisfied by both the project and the supplier.

- SP 2.1-1: Review candidate COTS products to ensure they satisfy the specified requirements that are covered under a supplier agreement.
- SP 2.2-1: Perform activities with the supplier as specified in the supplier agreement.
- SP 2.3-1: Ensure that the supplier agreement is satisfied before accepting the acquired product.
- SP 2.4-1: Transition the acquired products from the supplier to the project.

Since it is assumed that an off-the-shelf product will be used "as-is," the license for use of these products seldom provides any warranty for performance of the product. Ensuring satisfaction of the terms of the license and transition of the product tends to emphasize the right to use the product rather than the acceptability of the product for use in the solution. The acceptability of an off-the-shelf product and selection of its use in the solution is conducted as an engineering, not a project management, activity.

Integrated Project Management for IPPD

Purpose: Establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.

While there are no special considerations for the specific goals in this process area, the practices of this process area, in conjunction with the other IPPD process areas, are particularly important for the development of a COTS-based system. These practices provide the necessary foundation for both the generic spiral development and specific off-the-shelf product implications for defining and managing the stakeholder involvement necessary to support the development of a COTS-based system.

For a COTS-based system, it is particularly important to manage stakeholder involvement in identifying mismatches and conducting the necessary tradeoffs to resolve them. In the development of a COTS-based system relevant stakeholders include the end-user community, the system engineers, and the suppliers of the off-the-shelf products. Their involvement is significant and must be planned and managed.

SG 1: The project is conducted using a defined process that is tailored from the organization's set of standard processes.

- SP 1.1-1: Establish and maintain the project's defined process.
- SP 1.2-1: Use the organizational process assets and measurement repository for estimating and planning the project's activities.
- SP 1.3-1: Integrate the project plan and the other plans that affect the project to describe the project's defined process.
- SP 1.4-1: Manage the project using the project plan, the other plans that affect the project, and the project's defined process.
- SP 1.5-1: Contribute work products, measures, and documented experiences to the organizational process assets.

SG 2: Coordination and collaboration of the project with relevant stakeholders is conducted.

- SP 2.1-1: Manage the involvement of the relevant stakeholders in the project.
- SP 2.2-1: Participate with relevant stakeholders to identify, negotiate, and track critical dependencies.
- SP 2.3-1: Resolve issues with relevant stakeholders.
- SP 2.4-1: Transition the acquired products from the supplier to the project.

SG 3: The project is conducted using the project's shared vision.

- SP 3.1-1: Identify expectations, constraints, interfaces, and operational conditions applicable to the project's shared vision.
- SP 3.2-1 Establish and maintain a shared vision for the project.

SG 4: The integrated teams needed to execute the project are identified, defined, structured, and tasked.

- SP 4.1-1: Determine the integrated team structure that will best meet the project objectives and constraints.
- SP 4.2-1: Develop a preliminary distribution of requirements, responsibilities, authorities, tasks, and interfaces to teams in the selected integrated team structure.
- SP 4.3-1: Establish and maintain teams in the integrated team structure.

There are no special considerations for these specific goals.

Risk Management

Purpose: Identify potential problems before they occur, so that risk-handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.

Risk Management practices are essential to the successful management of the spiral development approach required for a COTS-based system—even though some organizations will have less discipline or formality in their implementation of the specific practices. Risk management should not be considered an independent activity; it must be integrated with project-planning activities as well as the engineering activities.

SG 1: Preparation for risk management is conducted.

SP 1.1-1: Determine risk sources and categories.

SP 1.2-1: Define the parameters used to analyze and categorize risks, and the parameters used to control the risk management effort.

SP 1.3-1: Establish and maintain the strategy to be used for risk management.

Risk management for a COTS-based system must include all aspects of building, fielding, and supporting the solution. Of special note in such systems is that, in many cases, the highest source of risk to the success of the project is the readiness and willingness, or lack thereof, of the targeted end users to accept and use the designed solution.

SG 2: Risks are identified and analyzed to determine their relative importance.

SP 2.1-1: Identify and document the risks.

SP 2.2-1: Evaluate and categorize each identified risk using the defined risk categories and parameters, and determine its relative priority.

Project management risks, technical risks (including risks such as the ability of a given off-the-shelf product to scale to meet project needs or interoperate with other critical products of the solution), and process risks are identified and analyzed continuously throughout the life of the project to determine their impact on both the project and the evolving definition of the solution.

SG 3: Risks are handled and mitigated, where appropriate, to reduce adverse impacts on achieving objectives.

SP 3.1-1: Develop a risk mitigation plan for the most important risks to the project, as defined by the risk management strategy.

SP 3.2-1: Monitor the status of each risk periodically and implement the risk mitigation plan as appropriate.

In spiral development, high-priority project risks are used directly to determine the objectives for each iteration. The project plan is updated in each iteration to reflect mitigated and new risks.

Integrated Teaming

Purpose: Form and sustain an integrated team for the development of work products.

The practices in this process area, with **Integrated Project Management for IPPD** and **Organizational Environment for Integration** practices, provide the necessary foundation to form and sustain empowered teams. There are no special considerations for this process area; however, it is a particularly important process area for the development of a COTS-based system. It is strongly recommended that the organization use practices from this process area as it plans for and develops the solution to facilitate direct involvement of a diverse set of stakeholders in project teams.

SG 1: A team composition that provides the knowledge and skills required to deliver the team's product is established and maintained.

SP 1.1-1: Identify and define the team's specific internal tasks to generate the team's expected output.

SP 1.2-1: Identify the knowledge, skills, and functional expertise needed to perform team tasks.

SP 1.3-1: Assign the appropriate personnel to be team members based on required knowledge and skills.

There are no special considerations for this specific goal.

SG 2: Operation of the integrated team is governed according to established principles.

SP 2.1-1: Establish and maintain a shared vision for the integrated team that is aligned with any overarching or higher level vision.

SP 2.2-1: Establish and maintain a team charter based on the integrated team's shared vision and overall team objectives.

SP 2.3-1: Clearly define and maintain each team member's roles and responsibilities.

SP 2.4-1: Establish and maintain integrated team operating procedures.

SP 2.5-1: Establish and maintain collaboration among interfacing teams.

The development of a COTS-based system demands significant stakeholder involvement because an off-the-shelf product may require extensive enterprise process engineering. This specific goal provides, at the team level, support for collaboration among diverse team members to effectively develop a solution that meets the needs of the enterprise. Understanding the needed composition of the integrated team, and using techniques to quickly involve diverse stakeholders and sustain that involvement throughout the life of the solution, is key.

Integrated Supplier Management

Purpose: Proactively identify sources of products that may be used to satisfy the project's requirements and to manage selected suppliers while maintaining a cooperative project-supplier relationship.

The specific practices of this process area are critical to a COTS-based system. However, they must be reoriented to support the proactive identification of off-the-shelf product sources and to maintain a cooperative relationship between the project and its suppliers. These reoriented practices must support the collection and maintenance of current marketplace information for the simultaneous definition of and trades among the various spheres of influence throughout the life of the solution.

SG 1: Potential sources of products that best fit the needs of the project are identified, analyzed, and selected.

SP 1.1-1: Identify and analyze potential sources of products that may be used to satisfy the project's requirements.

SP 1.2-1: Use a formal evaluation process to determine which sources of custom-made and off-the-shelf product to use.

The emphasis in this specific goal must change from the project management activity of selecting suppliers to an engineering activity that evaluates and selects products. In a COTS-based system the marketplace must be monitored continuously so that the project has as much warning as possible of changes that might affect either the project or the solution. In addition, monitoring the marketplace for potential new sources will require new practices that track the behavior of the marketplace. The relative importance of the product to the project and the project to the supplier will determine the opportunities for influencing market directions.

SG 2: Work is coordinated with suppliers to ensure the supplier agreement is executed appropriately.

SP 2.1-1: Monitor and analyze selected processes used by the supplier.

SP 2.2-1: For custom-made products, evaluate selected supplier work products.

SP 2.3-1: Revise the supplier agreement or relationship, as appropriate, to reflect changes in conditions.

Suppliers of off-the-shelf products will seldom react as suppliers of custom products do. Where suppliers of custom products will build products to a project's specific needs, suppliers of off-the-shelf products do not. More importantly, suppliers of off-the-shelf products will not generally allow monitoring of the processes or work products they use to build their product unless they are commonly considered part of the product. As a result, practices such as SP2.1-1, may not be applicable for COTS-based system processes. Instead, an experimental facility that can be used to evaluate new and changed products is essential for project success. This facility should represent the operational environment as closely as possible and support evaluation of enterprise process impacts as well as engineering assessment of the suitability of each product, product enhancement, or product patch.

Quantitative Project Management

Purpose: Quantitatively manage the project's defined process to achieve the project's established quality and process-performance objectives.

This process area has equivalent value to custom software development and COTS-based system projects. The organization will find, however, that the subprocesses selected and the data used for quantitative project management may be substantially different for a COTS-based system project.

SG 1: The project is quantitatively managed using quality and process-performance objectives.

- SP 1.1-1: Establish and maintain the project's quality and process-performance objectives.
- SP 1.2-1: Select the subprocesses that compose the project's defined process based on historical stability and capability data.
- SP 1.3-1: Select the subprocesses of the project's defined process that will be statistically managed.
- SP 1.4-1: Monitor the project to determine whether the project's objectives for quality and process performance will be satisfied, and identify corrective action as appropriate.

SG 2: The performance of selected subprocesses within the project's defined process is statistically managed.

- SP 2.1-1: Select the measures and analytic techniques to be used in statistically managing the selected subprocesses.
- SP 2.2-1: Establish and maintain an understanding of the variation of the selected subprocesses using the selected measures and analytic technique.
- SP 2.3-1: Monitor the performance of the selected subprocesses to determine their capability to satisfy their quality and process-performance objectives, and identify corrective action as necessary.
- SP 2.4-1: Record statistical and quality management data in the organization's measurement repository.

There are no special considerations for these specific process areas.

5.2 Engineering Process Areas

The CMMI engineering process areas cover the engineering activities related to development and maintenance activities that are shared across solution and software engineering disciplines. As with a custom development project, there is a critical need for discipline in these engineering activities for projects producing COTS-based systems. There are, however, some unique considerations for these activities defined for a COTS-based system. These considerations and their implications on the CMMI engineering process areas are summarized in Table 4 and will be discussed in the following paragraphs.

Table 4: Summary of Implications for Engineering Process Areas

Process Area	Interpretation
Requirements Management	Use as-is with an understanding that the processes must begin at project start and be robust enough to track and manage negotiated changes as understanding of the capabilities of the off-the-shelf products improves.
Requirements Development	Prioritize requirements and gain consensus on a minimum set of "must have" needs. Define requirements in enterprise terms to facilitate negotiations—expect that the requirements will evolve as more is learned about the off-the-shelf products and the shape of the solution. Form requirements based on hands-on knowledge of product capabilities.
Technical Solution	Form and maintain alternative solutions through simultaneous definition and trades. Build and maintain an experimentation facility for evaluation of off-the-shelf products in the context of the alternative solutions. Design the alternative solutions to support product evolution.
Product Integration	Use as-is to assemble integrated executable representations of the solution in every iteration.
Verification	Use as-is from the start of the project to verify that work products accurately reflect the state of the solution. Involve all stakeholders in peer reviews and verification of work products that affect enterprise processes.
Validation	Use as-is from the start of the project to validate that the solution provides useful capability to the enterprise. Involve all stakeholders to confirm that negotiation results are accurately reflected.

Requirements Management

Purpose: Manage the requirements of the project's products and product components and identify inconsistencies between those requirements and the project's plans and work products.

The practices in **Requirements Management** describe the activities needed for identifying and controlling changes to requirements and ensuring that affected plans and data are kept current. Robust requirements management processes, techniques, and tools must be defined and practiced from the *beginning* of a COTS-based system project. As end users come to better understand off-the-shelf product capabilities, as the products change with new upgrades, or as new products or technology become available, existing requirements are subject to change. The negotiation and tradeoffs of mismatches between requirements and products starts with project initiation and continues until the solution is retired.

SG 1: Requirements are managed and inconsistencies with project plans and work products are identified.

SP 1.1-1: Develop an understanding with the requirements providers on the meaning of the requirements.

SP 1.2-2: Obtain commitment to the requirements from the project participants.

SP 1.3-1: Manage changes to the requirements as they evolve during the project.

SP 1.4-2: Maintain bidirectional traceability among the requirements and the project plans and work products.

SP 1.5-1: Identify inconsistencies between the project plans and work products and the requirements.

Many programs believe that to "manage" requirements, they need to "control" them. For COTS-based systems, managing requirements does not imply that requirements should not change. Rather, there will be continual change as the end users discover and negotiate requirements based on improved understanding of off-the-shelf product capabilities. Still, changes to the requirements must be deliberately and carefully managed so that commitments can be adjusted.

Requirements Development

Purpose: Produce and analyze customer, product, and product-component requirements.

The identification and analysis of customer needs and formation into requirements is just as important a practice for COTS-based systems as they are for other types of systems. However, the practices defined in this process area must be performed simultaneously (with tradeoffs) with the practices of the **Technical Solution**, **Product Integration**, **Verification**, and **Validation** process areas—and with practices that engineer enterprise processes. These activities must be structured such that the affected stakeholders have hands-on interaction with the candidate off-the-shelf products. Requirements will change as the various stakeholders better understand the opportunities and limitations of the off-the-shelf offerings.

SG 1: Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements.
SP 1.1-1 Identify and collect stakeholder needs, expectations, constraints, and interfaces for all phases of the product life cycle.
SP 1.1-2 Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product life cycle.
SP 1.2-1 Transform stakeholder needs, expectations, constraints, and interfaces into customer requirements.

COTS-based system requirements must be formed through simultaneous definition and tradeoff among all spheres of influence. This means that requirements are balanced among what stakeholders want, what is available off-the-shelf, what the engineers can assemble, and what meets project management constraints. Therefore

- Requirements are stated in terms of enterprise needs, and prioritized to support ongoing negotiations as the definition of the solution evolves.
- Formation of requirements is strongly influenced by hands-on knowledge of the capabilities of the off-the-shelf products.
- Enterprise processes and the associated requirements are negotiated to better align with the products.

SG 2: Customer requirements are refined and elaborated to develop product and product-component requirements.
SP 2.1-1: Establish and maintain product and product-component requirements, which are based on the customer requirements.
SP 2.2-1: Allocate the requirements for each product component.
SP 2.3-1: Identify interface requirements.

A COTS-based system project must include processes that identify and gain consensus among affected stakeholders on the minimum set of "must have" stakeholder needs, architecturally significant requirements, and quality attributes (e.g., evolvability, reliability, performance thresholds). Prioritizing the requirements as described will help development activities focus on the most critical needs early, create a shared understanding of which requirements can be modified to allow the use of an off-the-shelf product, and provide a basis for ongoing negotiations among the affected stakeholders.

SG 3: The requirements are analyzed and validated, and a definition of required functionality is developed.
SP 3.1-1: Establish and maintain operational concepts and associated scenarios.
SP 3.2-1: Establish and maintain a definition of required functionality.
SP 3.3-1: Analyze requirements to ensure that they are necessary and sufficient.
SP 3.4-3: Analyze requirements to balance stakeholder needs and constraints.
SP 3.5-1: Validate requirements to ensure the resulting product will perform appropriately in its intended-use environment.
SP 3.5-2: Validate requirements to ensure the resulting product will perform as intended in the user's environment using multiple techniques as appropriate.

Use cases and operational scenarios are particularly valuable practices for COTS-based systems. Their use allows affected stakeholders to model and reason about the current and target enterprise processes and the implications of candidate products to those processes. This will facilitate more reasoned tradeoffs between requirements, enterprise processes, and available products.

Technical Solution

Purpose: Design, develop, and implement solutions to requirements. Solutions, designs, and implementations encompass products, product components, and product-related life-cycle processes either singly or in combinations as appropriate.

Technical Solution practices provide the mechanisms necessary to design, develop, and implement solutions to requirements for a COTS-based system. Discipline in the processes that form the architecture and design for COTS-based systems is often more critical than is the case for custom developed solutions. Effectively creating and evolving a COTS-based system requires far more than "simply" selecting one or more off-the-shelf products. All aspects of the solution are affected by the decision to use one or more products and the focus must be on engineering the solution as a whole. Where in custom developed solutions, developers or maintainers can "adjust" the code to compensate for poor design choices, in a COTS-based system the project's developers or maintainers do not typically have access to the code.

SG 1: Product or product-component solutions are selected from alternative solutions.
SP 1.1-1: Develop alternative solutions and selection criteria.
SP 1.1-2: Develop detailed alternative solutions and selection criteria.
SP 1.2-2: Evolve the operational concept, scenarios, and environments to describe the conditions, operating modes, and operating states specific to each product component.
SP 1.3-1: Select the product-component solutions that best satisfy the criteria established.

Different alternative solutions may meet the same general need but contain different off-the-shelf products, satisfy different requirements, demand very different architectures, and imply different enterprise processes. All affected stakeholders must be involved in evaluating and selecting among COTS-based system alternatives. Buying into a pre-determined view of the solution should be avoided. Selecting alternatives is inextricably linked with selection of off-the-shelf products that are included in that alternative.

Hands-on experimentation with each alternative will be necessary—preferably in an environment that reflects the operational environment. Information on individual off-the-shelf products from marketing brochures or supplier websites is insufficient. The stakeholders must see candidate products in the way that they will be used in an actual work environment.

Given the volatility of off-the-shelf products, previous decisions may need to be revisited when a product changes or a new product becomes available. Alternative architectures must be developed, analyzed, and maintained throughout the life of the solution so organizations can react to changes in the products with minimal disruption.

A word of extreme caution: modification of the off-the-shelf products is a sizeable risk to a project and should not be entered into readily. Specific practice 1.1-1 states: "COTS alternatives may be used with or without modification. Sometimes such items may require modifications to aspects such as interfaces or a customization of some of the features to better achieve product requirements." Any necessary modifications to COTS products, even "tailoring" or "customization," pose significant risk to the long-term maintenance of a COTS-based system. The maintenance implications of any adjustments to an off-the-shelf product should be carefully considered. These adjustments will have to be accommodated with each new release of the product—or the new release of a product that interfaces with it. If the product must be adjusted for use in the solution, this should be done either via mechanisms provided by the product supplier or through "wrappers" that protect the integrity of the product. In addition, beware of any use of an off-the-shelf product that is different from that intended by the supplier—unintended use may mean that future releases of the product will not meet project needs.

SG 2: Product or product-component designs are developed.
SP 2.1-1: Develop a design for the product or product component.
SP 2.2-3: Establish and maintain a technical data package.
SP 2.3-1: Establish and maintain the solution for product-component interfaces.
SP 2.3-3: Design comprehensive product-component interfaces in terms of established and maintained criteria.
SP 2.4-3: Evaluate whether the product components should be developed, purchased, or reused based on established criteria.

While custom products are *created,* COTS-based systems are *composed* from available products. Some have interpreted this specific goal to mean that solutions are designed based on a top-down analysis and decomposition, however, COTS-based systems must be designed "bottom up" as off-the-shelf products are evaluated alone, and in combination with other products and elements of the solution, in the context of the evolving definition of alternative solutions.

An integral part of each alternative solution is a product integration strategy. This strategy describes where "wrappers" and "glue" are needed to link products within the solution, what project standards or protocols will be used in the solution, and how the wrappers or glue will be maintained as the products change. A different product integration strategy may be needed for each alternative. The strategy will need to be assessed and updated as each product changes—for the life of the solution.

SG 3: Product components, and associated support documentation, are implemented from their designs.
SP 3.1-1 Implement the designs of the product components.
SP 3.2-1 Develop and maintain the end-use documentation.

For a COTS-based system, any custom products and any "wrappers" or "glue" necessary to link among products are implemented using the practices of this specific goal. Any wrappers or glue that are contained in the delivered solution will have to be maintained throughout the life of the solution as new versions of the off-the-shelf products are released.

Product Integration

Purpose: Assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product.

These practices should be used as-is. However, for COTS-based systems, **Product Integration** practices are needed in each iteration to develop executable representations of the alternate solutions that are adequate to demonstrate consensus among the stakeholders on definition of the solution and progress in meeting project objectives.

Building executable representations of the agreed-upon solution as it evolves (i.e., in each iteration) is an essential activity in a COTS-based system development. The form of the executable representation will vary from a mock-up of critical stakeholder needs in early iterations, to an evolutionary prototype that reflects the architecture and evolves to become the fielded solution in later iterations. Later executable representations must test the necessary infrastructure and any other systems with which the solution must interact and the enterprise processes affected by the solution.

SG 1: Preparation for product integration is conducted.
SP 1.1-1: Determine the product-component integration sequence.
SP 1.2-2: Establish and maintain the environment needed to support the integration of the product components.
SP 1.3-3: Establish and maintain procedures and criteria for integration of the product components.

SG 2: The product-component interfaces, both internal and external, are compatible.
SP 2.1-1: Review interface descriptions for coverage and completeness.
SP 2.2-1: Manage internal and external interface definitions, designs, and changes for products and product components.

SG 3: Verified product components are assembled and the integrated, verified, and validated product is delivered.
SP 3.1-1: Confirm, prior to assembly, that each product component required to assemble the product has been properly identified, functions according to its description, and that the product-component interfaces comply with the interface descriptions.
SP 3.2-1: Assemble product components according to the product integration sequence and available procedures.
SP 3.3-1: Evaluate assembled product components for interface compatibility.
SP 3.4-1: Package the assembled product or product component and deliver it to the appropriate customer.

There are no special considerations for these specific goals.

Verification

Purpose: Ensure that selected work products meet their specified requirements.

Verification practices should be used as-is. However, for a COTS-based system development, verification practices are used in every iteration from the early discovery stages of a COTS-based development to verify that the artifacts are built correctly and accurately reflect what is known about the solution at that point in time. The level of rigor applied will vary depending on the stage of a project and the level of detail reflected in the artifacts. In the early iterations of COTS-based systems projects, these practices will be applied in an informal setting—the rigor will increase as the design evolves and the artifacts become increasingly detailed over subsequent iterations. Because all of the artifacts are being developed simultaneously, all affected stakeholders must be involved in peer reviews and other verification activities.

Verification should not be confused with product evaluation, which examines the suitability of each off-the-shelf product for use in the solution. In most cases, a separate (from COTS product evaluation) integration capability will be required. The facility used for verification will demand strict configuration control to support the high repeatability demanded by the continuous testing required in a spiral development approach as the solution definition evolves and responds to changes in off-the-shelf products.

SG 1: Preparation for verification is conducted.
SP 1.1-1: Select the work products to be verified and the verification methods that will be used for each.
SP 1.2-2: Establish and maintain the environment needed to support verification.
SP 1.3-3: Establish and maintain verification procedures and criteria for the selected work products.

SG 2: Peer reviews are performed on selected work products.
SP 2.1-1: Prepare for peer reviews of selected work products.
SP 2.2-1: Conduct peer reviews on selected work products and identify issues resulting from the peer review.
SP 2.3-2: Analyze data about preparation, conducting, and results of the peer reviews.

SG 3: Selected work products are verified against their specified requirements.
SP 3.1-1: Perform verification on the selected work products.
SP 3.2-2: Analyze the results of all verification activities and identify corrective action.

There are no special considerations for these specific goals.

Validation

Purpose: Demonstrate that a product or product component fulfills its intended use when placed in its intended environment.

Validation practices should be used as-is. However, for a COTS-based system development, validation practices are used in every iteration from the early discovery stages of a COTS-based development to validate that the solution, as it is currently understood, defines a useful capability that provides value to the enterprise. The level of rigor applied will vary depending on the stage of a project and the level of detail in what is known about the solution. In the early iterations of COTS-based systems projects, these practices will be applied in an informal setting—the rigor will increase as the design evolves.

While the process area indicates that "often" end users are involved, for COTS-based systems, the end users must *always* be involved in validation to confirm the results of any and all negotiations. Given the demands of spiral development, a required element for COTS-based systems, selected intermediate work products will require frequent re-validation. The executable representations formed in each iteration are essential to validate the evolving definition of the solution. This includes any early mock-ups or prototypes.

SG 1: Preparation for validation is conducted.
SP 1.1-1: Select products and product components to be validated and the validation methods that will be used for each.
SP 1.2-2: Establish and maintain the environment needed to support validation.
SP 1.3-3: Establish and maintain procedures and criteria for validation.

SG 2: The product or product components are validated to ensure that they are suitable for use in their intended operating environment.
SP 2.1-1: Perform validation on the selected products and product components.
SP 2.2-1: Analyze the results of the validation activities and identify issues.

There are no special considerations for these specific goals.

5.3 Support Process Areas

The CMMI support process areas address processes that are used in the context of performing other processes. In general the Support process areas address processes that are targeted towards the project, and may address processes that apply more generally to the organization. For example, **Process and Product Quality Assurance** can be used with all the process areas to provide an objective evaluation of the processes and work products described in all of the process areas. As with a custom development project, there is a critical need for discipline in these support activities for projects producing COTS-based systems. There are, however, some unique considerations for these activities defined for a COTS-based system. These considerations and their implications on the CMMI support process areas are summarized in Table 5 and will be discussed in the following paragraphs.

Table 5: Summary Implications for Support Process Areas

Process Area	Interpretation
Configuration Management	Use as-is—make robust enough to accommodate spiral development and product volatility. Establish baselines early and baseline additional off-the-shelf attributes.
Process and Product Quality Assurance	Use as-is—evaluate quality iteratively as the solution evolves. Evaluate off-the-shelf product quality in the context of the solution.
Measurement and Analysis	Use as-is—different measurements and different interpretation of measures may be needed.
Decision Analysis and Resolution	Use as-is—identify who makes what decisions and who will participate in support of simultaneous definition and tradeoffs.
Organizational Environment for Integration	Use as-is—use as a foundation to form and sustain stakeholder teams.
Causal Analysis and Resolution	Use as-is—employ new analysis techniques and methods.

Configuration Management

Purpose: Establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

Configuration Management practices should be used as-is. However, for a COTS-based system, the configuration management and change management system must be robust enough to accommodate configuration items that change often. In addition, it must start early in the program to capture the work products associated with early evaluation of off-the-shelf products and to accommodate the concurrent development of work products throughout multiple iterations that is an essential element of a spiral development approach.

SG 1: Baselines of identified work products are established.
SP 1.1-1: Identify the configuration items, components, and related work products that will be placed under configuration management.
SP 1.2-1: Establish and maintain a configuration management and change management system for controlling work products.
SP 1.3-1: Create or release baselines for internal use and for delivery to the customer.

In addition to the work products associated with a custom solution, the configuration baseline for a COTS-based system should include how each product is used in the solution, which product versions are used in each release of the solution, license information, product patches, and dependencies among products.

SG 2: Changes to the work products under configuration management are tracked and controlled.
SP 2.1-1 Track change requests for the configuration items.
SP 2.2-1 Control changes to the configuration items.

The change management system must be robust and efficient enough to handle the high level of activity that can be expected from updates to most work products in each iteration and the asynchronous releases of off-the-shelf products. These releases may be asynchronous with other products and/or asynchronous with releases of the solution.

In addition, it is likely that there will be cascading dependencies among off-the-shelf products—products that are dependent on one another in such a way that a change or upgrade of one is likely to force a corresponding change or upgrade in one or more others.

SG 3: Integrity of baselines is established and maintained.
SP 3.1-1 Establish and maintain records describing configuration items.
SP 3.2-1 Perform configuration audits to maintain integrity of the configuration baselines.

It is important that each iteration begins with correctly configured work products and that the integrity of the output of the iteration is ensured.

Process and Product Quality Assurance

Purpose: Provide staff and management with objective insight into processes and associated work products.

Process and Product Quality Assurance practices should be used as-is. However, for a COTS-based system quality assurance processes will need to accommodate a spiral development approach. Quality assurance must evaluate quality in each iteration as each of the work products evolves.

SG 1: Adherence of the performed process and associated work products and services to applicable process descriptions, standards, and procedures is objectively evaluated.

SP 1.1-1: Objectively evaluate the designated performed processes against the applicable process descriptions, standards, and procedures.

SP 1.2-1: Objectively evaluate the designated work products and services against the applicable process descriptions, standards, and procedures.

The project must expect to treat an off-the-shelf product as a "black box." The project will seldom be allowed visibility into the work processes or supporting work products of the supplier of an off-the-shelf product. The quality of an off-the-shelf product is more appropriately measured in terms that characterize how the product affects the desired quality attributes of the solution. This will require substantially different techniques and tools. In addition to the evaluating the processes used to manage the project, therefore, the focus in quality assurance will be on the processes used to evaluate, select, and integrate the products in the context of the evolving definition of the solution.

Note: Any glue, wrappers, or parameter tables used to configure the product can be treated as custom developed products.

SG 2: Noncompliance issues are objectively tracked and communicated, and resolution is ensured.

SP 2.1-1: Communicate quality issues and ensure resolution of noncompliance issues with the staff and managers.

SP 2.2-1: Establish and maintain records of the quality assurance activities.

When an off-the-shelf product is found to be in some way noncompliant with the solution, the project may be limited in its choices for resolution. The project should track and communicate the issue; however, the supplier may or may not resolve the issue to the satisfaction of the project. In this case, the project can choose another product, compensate for the noncompliance with additional custom development, or waive the desired solution compliance.

Measurement and Analysis

Purpose: Develop and sustain a measurement capability that is used to support management information needs.

Measurement and Analysis practices should be used as-is. The need for developing and sustaining a measurement and analysis capability for a COTS-based system is not different from that for a custom development project, although the actual measurements taken and the analysis and interpretation of the data may be different. Monitoring and controlling the requirements changes for a COTS-based system may require different measures or different interpretations of existing measures. For example, a COTS-based system project should expect any measures of requirements volatility values to be higher than for a custom software development, as the requirements are negotiated and traded off based on increasing knowledge of off-the-shelf products.

SG 1: Measurement objectives and activities are aligned with identified information needs and objectives.

- SP 1.1-1: Establish and maintain measurement objectives that are derived from identified information needs and objectives.
- SP 1.2-1: Specify measures to address the measurement objectives.
- SP 1.3-1: Specify how measurement data will be obtained and stored.
- SP 1.4-1: Specify how measurement data will be analyzed and reported.

SG 2: Measurement results that address identified information needs and objectives are provided.

- SP 2.1-1: Obtain specified measurement data.
- SP 2.2-1: Analyze and interpret measurement data.
- SP 2.3-1: Manage and store measurement data, measurement specifications, and analysis results.
- SP 2.4-1: Report results of measurement and analysis activities to all relevant stakeholders.

There are no special considerations for these specific goals.

Decision Analysis and Resolution

Purpose: Analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.

Decision Analysis and Resolution practices should be used as-is. However, in a COTS-based system, decision analysis and resolution is particularly important in supporting the negotiation and tradeoffs among the four spheres of influence. The project must define very early how decisions will be made, who will make the decisions, and what roles key stakeholders will play. There needs to be an understanding of the various levels of decision-making processes that will be needed, who has authority to make decisions at each level, and who will participate in the different decision-making processes. It is important that all the affected stakeholders are knowledgeable of how decisions will be made and how they can and should participate.

SG 1: Decisions are based on an evaluation of alternatives using established criteria.

- SP 1.1-1: Establish and maintain guidelines to determine which issues are subject to a formal evaluation process.
- SP 1.2-1: Establish and maintain the criteria for evaluating alternatives, and the relative ranking of these criteria.
- SP 1.3-1: Identify alternative solutions to address issues.
- SP 1.4-1: Select the evaluation methods.
- SP 1.5-1: Evaluate alternative solutions using the established criteria and methods.
- SP 1.6-1: Select solutions from the alternatives based on the evaluation criteria.

There are no special considerations for this specific goal.

Organizational Environment for Integration

Purpose: Provide an Integrated Product and Process Development (IPPD) infrastructure and manage people for integration.

There are no special considerations for this process area. However, **Organizational Environment for Integration** practices, with those of the **Integrated Project Management for IPPD** and **Integrated Teaming** process areas, provide the necessary foundation to form and sustain empowered teams necessary to support development and maintenance of a COTS-based system.

SG 1: An infrastructure that maximizes the productivity of people and affects the collaboration necessary for integration is provided.

- SP 1.1-1: Establish and maintain a shared vision for the organization.
- SP 1.2-1: Establish and maintain an integrated work environment that supports IPPD by enabling collaboration and concurrent development.
- SP 1.3-1: Identify the unique skills needed to support the IPPD environment.

SG 2: People are managed to nurture the integrative and collaborative behaviors of an IPPD environment.

- SP 2.1-1: Establish and maintain leadership mechanisms to enable timely collaboration.
- SP 2.2-1: Establish and maintain incentives for adopting and demonstrating integrative and collaborative behaviors at all levels of the organization.
- SP 2.3-1: Establish and maintain organizational guidelines to balance team and home organization responsibilities.

There are no special considerations for these specific goals.

Causal Analysis and Resolution

Purpose: Identify causes of defects and other problems and take action to prevent them from occurring in the future.

Causal Analysis and Resolution practices should be used as-is. However, for COTS-based systems, the techniques and methods used to identify the causes of defects and other problems may be different from those used in a custom software system. In particular, diagnostic techniques will be needed to isolate the source of subtle interaction problems that can arise as multiple off-the-shelf products—each of which operates as designed—are integrated in the solution.

SG 1: Root causes of defects and other problems are systematically determined.

- SP 1.1-1: Select the defects and other problems for analysis.
- SP 1.2-1: Perform causal analysis of selected defects and other problems and propose actions to address them.

SG 2: Root causes of defects and other problems are systematically addressed to prevent their future occurrence.

- SP 2.1-1: Implement the selected action proposals that were developed in causal analysis.
- SP 2.2-1: Evaluate the effect of changes on process performance.
- SP 2.3-1: Record causal analysis and resolution data for use across the project and organization.

There are no special considerations for these specific goals.

6 Summary

Using off-the-shelf products to meet the needs of business, mission, or operational applications demands a fundamental change in how solutions are engineered—and this change has implications on engineering, management, and enterprise processes. Building solutions using COTS products requires changed roles and responsibilities, new skills, and different processes. The processes for developing and maintaining COTS-based systems require more discipline than is often practiced in custom development projects, to accommodate the potentially volatile changes in the off-the-shelf products—changes over which the project has little or no control.

This analysis suggests that, while interpretation is required for those who wish to develop and maintain COTS-based systems, CMMI provides a sound basis for COTS-based system process improvement. Further, selecting off-the-shelf products and managing appropriate relationships with their suppliers is more than just applying the Supplier Sourcing discipline within CMMI:

1. All four CMMI disciplines—Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing—must be used together for COTS-based system processes.
2. All of the CMMI process area categories—Project Management, Engineering, Process Management, and Support—are important to a COTS-based system project.
3. CMMI Process areas in the Project Management, Engineering, and Support categories will need to be integrated and applied differently to accommodate the process demands of COTS-based systems:
    - Many organizations will find that one of the biggest challenges in implementing processes for COTS-based systems is the transition to a risk-based spiral development approach. A spiral development approach facilitates the formation of requirements and the definition of the solution based on a clear understanding of the opportunities and limitations of the available off-the-shelf products. Implementing a spiral development approach requires that the engineering and program management practices be integrated to support the simultaneous definition and trades among the four spheres of influence and that artifacts that reflect the current understanding of the solution be verified and validated through frequent executable representations.

- Suppliers of off-the-shelf products will seldom respond to project needs or direction as a developer of custom software does. Effective processes will be needed to monitor the marketplace continuously across the life of the solution, for indicators of potential changes to the off-the-shelf products that could affect the solution and for opportunities to influence those changes.

4. While no specific practices exist in CMMI to explicitly address the activities that plan, design, and deploy any necessary changes to the enterprise processes, these activities are integral to developing and maintaining a COTS-based system. Integrating these activities demands that all affected end-users, who may represent very diverse interests, must be directly and actively included with the other stakeholders involved in engineering as well as management activities throughout the life of the solution.

References/Bibliography

All URLs are current as of this document's publication date.

[Albert 02] Albert, C. & Brownsword, L. *Evolutionary Process for Integrating COTS-Based Systems (EPIC)* (CMU/SEI-2002-TR-005, ADA408653). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. <http://www.sei.cmu.edu/publications/documents/02.reports/02tr005.html> (2002).

[Boehm 00] Boehm, B., edited by Hansen, W. J. *Spiral Development: Experience, Principles, and Refinements.* (CMU/SEI-2000-SR-008, ADA382590). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. http://www.sei.cmu.edu/publications/documents/00.reports/00sr008.html> (2000).

[Brownsword 98] Brownsword, L.; Carney, D.; & Oberndorf, P. "The Opportunities and Complexities of Applying COTS Components." *Crosstalk, 11*, 4 (April 1998): 4-6. <http://www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1998/04/applying.asp> (1998).

[Carney 00] Carney, D.; Hissam, S.; & Plakosh, D. "Complex COTS-based Software Systems: Practical Steps for their Maintenance." *Journal of Software Maintenance: Research and Practice*, *12*, 6 (2000): 357-376.

[Carney 03] Carney, D.; Oberndorf, P.; & Place, P. C. *A Basis for an Assembly Process for COTS-Based Systems (APCS)* (CMU/SEI-2003-TR-010). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/publications/documents/03.reports/03tr010.html> (2003).

[CBS 02] COTS-Based Systems Lessons Learned Web Pages. Software Engineering Institute. Carnegie Mellon University. <http://www.sei.cmu.edu/cbs/lessons> (2002).

[CMMI 02] CMMI Product Team: *Capability Maturity Model® Integration (CMMI®), Version 1.1: CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development and Supplier Sourcing: Continuous Representation* (CMMI-SE/SW/IPPD/SS, V1.1) (CMU/SEI-2002-TR-11, ADA339818). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. <http://www.sei.cmu.edu/publications/documents/02.reports /02tr011.html> (2002).

[Goldenson 95] Goldenson, D. & Herbsleb, J. *After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors That Influence Success* (CMU/SEI-95-TR-009, ADA302225). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995. <http://www.sei.cmu.edu/publications/documents/95.reports /95.tr.009.html> (1995).

[Oberndorf 00] Oberndorf, P.; Brownsword, L.; & Sledge, C. *An Activity Framework for COTS-Based Systems* (CMU/SEI-2000-TR-010, ADA385347). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications/documents/00.reports /00tr010.html> (2000).

[OSD 00] Office of the Secretary of Defense, *Commercial Item Acquisition: Considerations and Lessons Learned.* <http://www.abm.rda.hq.navy.mil/ciacll.pdf> (2000).

[USAF 00] United States Air Force Science Advisory Board: *Report on Ensuring Successful Implementation of Commercial Items in Air Force Systems,* (SAB-TR-99-03). 2000.

[Wallnau 02] Wallnau, K. C.; Hissam, S. A.; & Seacord, R. C. *Building Systems from Commercial Components.* New York, NY: Addison-Wesley, SEI Series in Software Engineering, 2002.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE October 2003	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE
Interpreting Capability Maturity Model® Integration (CMMI®) for COTS-Based Systems

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
Barbara Tyson, Cecilia Albert, Lisa Brownsword

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2003-TR-022

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ESC-TR-2003-022

11. supplementary notes

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

Experience shows that engineering commercial off-the-shelf (COTS)-based systems requires fundamental changes from traditional engineering: adjusted roles and responsibilities, new skills, and different processes. Practitioners are often surprised to find that building and supporting COTS-based systems demands more, not less, discipline in their management and engineering practices.

Many organizations have derived benefit from process improvement using capability maturity models and want to apply them as they build COTS-based systems. In addition, organizations building COTS-based systems want to apply the Capability Maturity Model® Integration (CMMI®). This leads to the question, "How should CMMI be interpreted for organizations building, fielding, and supporting a COTS-based system?"

This report shows that developing and maintaining COTS-based systems is more than selecting products and managing vendor relationships and is, therefore, more than just applying the Supplier Sourcing discipline within CMMI. The four CMMI disciplines—Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing—require interpretation and must be used together to promote improvement of an organization's processes for developing and maintaining COTS-based systems. This report summarizes what makes COTS-based systems unique and provides high-level guidance for interpreting and using CMMI practices to facilitate appropriate processes for COTS-based systems.

14. SUBJECT TERMS
Capability Maturity Model Integration, CMMI, COTS-Based Systems, process areas, spiral based approach, off-the-shelf

15. NUMBER OF PAGES
66

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

® Capability Maturity Model and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.